


**REMARKS**

Claims 1-15 are pending in the application. Claims 1-11 are amended to improve grammatical presentation.

In the event any questions arise regarding this communication or the application in general, the Examiner is invited to contact Applicants' undersigned representative at the telephone number listed below.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

By:   
M. David Ream  
Registration No. 35,333

P. O. Box 1404  
Alexandria, Virginia 22313-1404  
(703) 836-6620

Date: 07 November 2001

RECEIVED  
NOV 09 2001  
Technology Center 2100



Attachment to Preliminary Amendment 07 November 2001

**Marked-up Claims**

1. A method for [visualizing] representing and processing a value assembly process, the value assembly process being [visualized] represented as a set of value assembly lines, which value assembly lines are arranged on a number of different hierarchical levels, the [value assembly lines having the following essential properties] method comprising:
  - [-] providing each value assembly line has precisely one output interface and at least one input interface;
  - [-] [the] each value assembly line [receives] receiving input value packages via the input interfaces;
  - [-] combining the input value packages [are combined] in [the] each value assembly line in accordance with rules defined in a specific main line function, making a value contribution [being made], and generating [the] a value added package [being generated];
  - [-] in each value assembly line, making the value added package [is made] available via the output interface; [and]
  - [-] wherein similar types of information are contained in the input value packages and in the value added package;[the process visualization having the following basic properties:]
  - [-] providing an uppermost hierarchy level [has] having precisely one value assembly line of the highest hierarchy level[, which generates a value added package];

the precisely one value assembly line generating a value added package;

**Attachment to Preliminary Amendment 07 November 2001**

**Marked-up Claims**

[-] the precisely one value assembly line of the highest hierarchy level [receives]  
receiving input value packages via its input interfaces from lower-order value assembly lines;

[-] [when] representing a lower-order value assembly line [is focused on, it is  
visualized] in an entirely similar way as a value assembly line which receives its input value  
packages from lower-order value assembly lines and whose value added package is provided  
as input value package for the value assembly line of the uppermost hierarchy level; and

[-] [on] for at least one of each hierarchy level down to a lowermost hierarchy level  
[it is possible in each case to focus on] representing a lower-order value assembly line of [this]  
the at least one hierarchy level[, which is visualized] in a similar way as a value assembly line  
of the next lower hierarchy level, which likewise receives input value packages via input  
interfaces, combines [these] received input value packages, makes a value contribution, and  
makes a value added package available at the output interface;

[in such a way that] representing the value assembly process [is visualized] as a fractal  
process in the case of which the structure of all value assembly lines is similar on all hierarchy  
levels, value packages being processed in accordance with the following steps:

[-] on a lowermost hierarchy level, supplying value packages [are supplied]  
to the value assembly lines of the lowermost hierarchy level across the system boundaries of  
the value assembly process [under consideration];

[-] combining the input value packages of a lowermost hierarchy level [are  
combined] in value assembly lines of [this] the lowermost hierarchy level in accordance with

**Attachment to Preliminary Amendment 07 November 2001**

**Marked-up Claims**

its main line function, increasing the value of the value packages [is increased] by a value contribution of the value assembly line and/or of the main line function, and making a value added package [is made] available at the output interface of the value assembly line;

[-] on all hierarchy levels up to a highest hierarchy level, passing the value added package [is passed] on to precisely one value assembly line of the next higher hierarchy level, [and serves this value assembly line] as an input value package;

[-] wherein the value flows take place strictly in one direction, in each case from a lower hierarchy level into a higher hierarchy level, and the value assembly lines of a hierarchy level are not interconnected.

2. The method as claimed in claim 1, comprising:

[in which] comparing each value added package [is compared] with a reference value added [package,] package;

and [in which] detecting and reporting impermissible deviations of the value added package and of the reference value added package [are detected and reported] via a warning function.

3. A computer program for simulating and illustrating a value assembly process, the process comprising a number of self-similar value assembly lines which are arranged on different hierarchy levels and are independent of one another on a hierarchy level, in which

**Attachment to Preliminary Amendment 07 November 2001**

**Marked-up Claims**

value assembly process value packages are transferred in each case from a value assembly line of a lower-order hierarchy level into a value assembly line of a higher-order hierarchy level, input value packages being combined in each value assembly line, a value contribution being made, and a value added package being generated, [which] the computer program [includes] comprising:

[-] machine-readable instruction sequences of a first, higher-order hierarchy level, which prompt a computer to read in data from at least one data form and to combine these data using specific combining rules, and to store the output data thus determined in a data form on a computer-readable storage medium and/or an output medium;

[-] mutually independent machine-readable instruction sequences of at least one further hierarchy level of lower order than the first hierarchy level, which instruction sequences prompt a computer to read out data at least from a number of data forms, to combine them with one another, and to store the results of combination in an output data form of this hierarchy level in each case; and

[-] machine-readable instruction sequences which prompt a computer to read data into input data forms from an input unit;

[in which] wherein the input data forms, which are being read on a specific hierarchy level during the program run, are output data forms of a hierarchy level of lower order than this hierarchy level, or input data [forms:] forms:

**Attachment to Preliminary Amendment 07 November 2001**

**Marked-up Claims**

[in which] wherein all data forms have a standardized data format in such a way that all output data forms which are generated during the run of instruction sequences on an arbitrary hierarchy level, and all input data forms have a common data structure, that is to say data which are of one information type are always stored in the same position in a form;

and [in which] wherein all different machine-readable instruction sequences of all different hierarchy levels are generated from identical source codes from which instructions for reading in the specific data forms and relating to the specific combinations are created by a specific parameter file either during generation of the sequences which can be executed, or during the running time of the computer program.

4. The computer program as claimed in claim 3, [in which] wherein the data contained in the forms constitute input value packages and value added packages of the value assembly process.

5. The computer program as claimed in claim 3, [in which] wherein the data generated by an instruction sequence are classified according to their qualitative information content and stored in different classes of standardized data forms.

6. The computer program as claimed in claim 3, [in which] wherein during execution of the computer program or individual sequences of the computer program the generated value

**Attachment to Preliminary Amendment 07 November 2001**

**Marked-up Claims**

added packages are compared with a reference value added package in each case, which reference value added package contains specification data, and [in which] wherein a report is made via a warning function in the event of impermissible specification deviations.

7. The computer program as claimed in claim 3, [in which] wherein machine-readable sequences of different value assembly subprocesses run on different computers, and data forms are transferred via long-distance data lines.

8. A method for [visualizing] visually representing a value assembly process on an output unit of a computer system, the process comprising a number of self-similar value assembly lines which are arranged on different hierarchy levels and are independent of one another on a hierarchy level, [in which value assembly process] wherein value packages are transferred in each case from a value assembly line of a lower-order hierarchy level into a value assembly line of a higher-order hierarchy level, input value packages [being] are combined in each value assembly line, a value contribution [being] is made, and a value added package [being] is generated, the computer system [including, inter alia,] comprising a central processing unit and a pointing device in addition to the output unit, [in which] the method comprising:

representing a value assembly line of one hierarchy level [is visualized in each case] on the output unit, said value assembly line being [visualized] represented as an arrow at the

**Attachment to Preliminary Amendment 07 November 2001**

**Marked-up Claims**

tip of which a value added package is transferred, and representing lower-order value assembly lines which are value assembly lines of a lower-order hierarchy level, [are likewise visualized] as arrows the tips of which are applied to the shaft of the arrow which [visualizes] represents the value assembly line of the higher-order hierarchy [level,] level;

[it being possible for the] a user of the computer system [to use the pointing device to select] selecting a lower-order value assembly line via the pointing device; [and to focus on the latter in such a way as to visualize it in the focused visualization]

representing the selected lower-order value assembly line in the same way as the value assembly line of the higher-order hierarchy level as an arrow with smaller arrows running up to it [, and it being possible for the user to use the pointing device to select the arrow tip of the value assembly line, whereupon the value assembly line of the higher-order hierarchy level is visualized in a similar way and such that the path of an integral part of the value assembly process firstly can be traced back without difficulty to its origin from a higher hierarchy level, and the contribution of this integral part to the overall value assembly process can be traced through all hierarchy levels].

9. The method as claimed in claim 8, [in which] wherein a computer program in accordance with claim 3 runs in the central processing unit of the computer system.



**Attachment to Preliminary Amendment 07 November 2001**

**Marked-up Claims**

10. The method as claimed in claim 8, [in which] wherein the arrows of value assembly processes which make different value contributions are [visualized] represented with the aid of different colors and/or line thicknesses.

11. The method as claimed in claim 8, in which a computer program runs in the central processing unit of the computer system, and in which the warning function prompts the computer system to display on the output unit an impermissible value deviation occurring on an arbitrary hierarchy level, and all arrows which [visualize] represent value assembly lines which are affected by the value deviation are [visualized] represented in a particularly emphatic type of [visualization] visual representation on the output unit in such a way that the impermissible deviation can be traced back immediately to its origin from a higher hierarchy level.